2007

First Read Rate Analysis of 2D-Barcodes for Camera Phone Applications as a Ubiquitous Computing Tool

Hiroko Kato
Edith Cowan University

Keng T. Tan
Edith Cowan University

10.1109/TENCON.2007.4428778
This is an Author's Accepted Manuscript of: Kato, H., & Tan, K. (2007). First read rate analysis of 2D-barcodes for camera phone applications as a ubiquitous computing tool. Proceedings of IEEE Tencon (IEEE Region 10 Conference). (pp. 1-4). Taipei, Taiwan. IEEE. Available here

First read rate analysis of 2D-barcodes for camera phone applications as a ubiquitous computing tool.

H. Kato and K.T. Tan
School of Computer Science and Information Science
Edith Cowan University
2 Bradford Street
Mount Lawley, WA 6050 AUSTRALIA

Abstract- **This paper presents a detailed study on the first read rate (FRR) of seven 2D-barcodes currently used for camera phone applications. The study revealed a few surprising observations. Through our analysis, we identified three key factors to improve the robustness of 2D-barcode reading, the range of the reading distance and the stability of the readers. This will contribute to the widespread use of 2D-barcode mobile technology as a ubiquitous computing tool.**

I. INTRODUCTION

Barcode is one of the most prevalent automatic identification, keyless data entry technologies. Traditional one dimensional (1D) barcode, which encodes limited number of globally unique digits,[1] works as an index to a backend database. It enables efficient sales and inventory management, providing real-time information on products. However, the demand for a barcode that carries more data in less space rose in certain industries such as pharmaceutical industries, which resulted in the invention of two dimensional (2D) barcodes. Carrying data within itself, 2D-barcode works as a portable data file.

In recent years, the combination of two mobile technologies, namely 2D-barcodes and camera phones, is gaining popularity as a promising ubiquitous computing tool. With the integration of the built-in cameras, mobile phones can work as scanners, barcode readers and portable data storages, maintaining network connectivity. When used together with such camera phones, 2D-barcodes work as a tag to connect the digital and real world. The most popular application is to link camera phones to Web pages via 2D-barcodes. Saved in mobile phones, 2D-barcodes can also be used as portable data files such as e-tickets/e-coupons. They can be purchased and exchanged via Internet. E-tickets shown on the phone display can be scanned and verified at the check-in counter or reception with no attendants, which results in faster ticket handling. Furthermore, no paper is used, making it environmentally friendly.

2D-barcodes offer a variety of convenient, fun and exciting applications. Nonetheless, this new mobile application area is still at its infancy. One reason is that this technology as a ubiquitous computing tool has not reached its maturity, which affects their stability, reliability and usability. To address this issue, we have embarked on a detailed study [1, 2] of all the currently available 2D-barcodes. We have examined each 2D-barcode system in terms of essential criteria for camera phone applications: data capacity, omni-directional symbol reading, error correction capability, support for low resolution cameras, reading robustness under different light conditions, legible distance of codes, security and multiple character-sets support.

In this paper, we present key factors that could enhance the robustness and usability of a 2D-barcode system[2] based on our analysis of the first read rates (FRRs) examination.

II. 2D-BARCODES FOR CAMERA PHONE APPLICATIONS

More than thirty different 2D-barcodes are currently in use. At present, seven 2D-barcodes are used for camera phone applications among them. These are: QR Code [3], VeriCode [4], Data Matrix [5], mCode [6], Visual Code [7], ShotCode [8] and ColorCode [9]. Fig. 1 presents these 2D-barcode symbols.[3]

The first four 2D-barcodes were invented to improve data capacity. In addition to their higher data capacities, they have various useful features that enhance the reading robustness of the codes such as error detection and correction capability. Hence, these 2D-barcodes can operate as robust, portable data files. We call them "database 2D-barcode" hereafter. With database 2D-barcodes, users can access the information they need at anytime, anywhere, regardless of network connectivity.

As for the last three 2D-Barcodes, they focus more on robust and reliable reading, taking into account the reading limitations of built-in cameras in mobile phones. They differ greatly from database 2D-barcodes in terms of data capacity. Each 2D-barcode basically works as an 'index' to link the digital and real world, relying on network connectivity to the Internet via mobile phones. We call them "index 2D-barcode", hereafter. The followings are brief description of each 2D-barcode, focusing the features that affect the robustness of reading.

A. *Database 2D-barcodes*

QR Code is capable of encoding all types of data including symbols, binary and multimedia and so forth. The maximum data capacity of numeric, alphanumeric and binary is 7,089 characters, 4,296 characters and 2,953 bytes, respectively. By applying Reed-Solomon error correction coding, up to 30% of original data can be restored even if the symbol is damaged. Masking technique and structured append feature [2] are also useful to enhance reading reliability of QR code.

VeriCode® and VSCode® share many features except for their data capacity. The maximum data capacity of VeriCode® is 500 bytes, whereas VSCode® can store up to 4,151 bytes. Using Reed-Solomon, a high percentage of Error Detection And Correction (EDAC) capability (ranging from 15% to 25%) of VeriCode®/ VSCode® enables the encoded data to be 100% restored even if up to 35% of the symbol is damaged.

[1] Some 1D-barcodes are able to encode alphanumeric and special characters.

[2] A 2D-barcode system includes 2D-barcode and its symbol, decoding software, and occasionally infrastructure to implement applications.

[3] ColorCode was included as a candidate since its symbol structure is two dimensional although it is sometimes referred to as a 3 dimensional code considering the use of color elements as another dimension.

1-4244-1272-2/07/$25.00

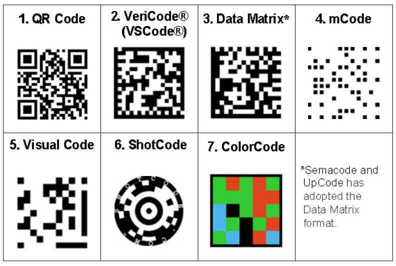

Figure 1. Examples of the 2.5 cm^2 sample symbols with an identical data.

Data Matrix uses two types of error correction algorithms, depending on the Error Checking and Correcting (ECC) level employed. Whereas ECC level 000 to 140 use convolutional code error correction, ECC-200 uses Reed-Solomon. The former offers five different error correction levels.[4] However, the correction level of ECC-200 is determined by the symbol size. Although ECC-000 to ECC-140 is still available, ECC-200 is now in common use. The maximum data capacity of Data Matrix is 3116 numeric digits, 2335 alphanumeric characters, or 1556 bytes. Features to enhance Data Matrix's reading robustness include its small symbol size, structured append function and data compaction [2].

Semacode [10] and UpCode [11] have adopted the Data Matrix format. The difference between Semacode and UpCode is that the former is used as a database code encoding plain-text URL, whereas the latter basically works as an index code.

mCode is specifically developed to meet the needs of emerging camera phone applications. mCode was designed to maximize the data capacity within a given space. Examples are mCode's dot (called blob) finder patterns and special compression form used for encoding URL. The largest code size is 44×44, which carries approximately 150 bytes of data. With Reed-Solomon coding, mCode uses variable size error correction polynomial, which reduces unnecessary bit waste.

B. *Index 2D-barcodes*

Visual code is the first 2D-barcode designed for camera phone applications from scratch. Visual Code System was developed to enable human-computer interactions using camera phones. Although the data capacity is limited (83 bits), Visual Code can function both as a portable database[5] and as an index to a remote database. For error detection and correction, Visual Code employs a (83, 76, 3) linear block code.

ShotCode stores an index consisting of 49 bits of data and provides links between the real and digital world, accessing remote databases. It was originally called SpotCode and was developed to enhance human-computer interactions [8]. SpotCode is a derivative of another circular 2D-barcode tag known as TRIP tag/code. TRIP code uses even parity check to detect possible decoding errors.

ColorCode has been used as an index-based code, encoding 10 digits. ColorCode includes a parity area that contains error parity check to detect any incorrect color recognition, which the system then corrects. The result of an exclusive operation (XOR) of code values in each column/row becomes the code value of the parity cell for the respective column/row, which is converted to its corresponding color value in the parity area of the symbol.

III. First Read Rate (FRR) Analysis

As a final step to our initial study presented in [1, 2], we have analyzed the first read rate (FRR) of each sampled 2D-barcode depicted in Fig. 1, where:

$$\text{First Read Rate (FRR)} = \frac{\text{Number of successful first reads}}{\text{Number of attempted first reads}}$$

The metric allows us to quantitatively verify the result obtained in our previous study [1, 2] and gauge the reading reliability of a given 2D-barcode. Furthermore, there is no published data on the FRR of the sampled 2D-barcodes.

For the analysis, we have created 4 samples for each sampled 2D-barcode (see Fig. 1). We encoded an identical set of data in the first 2 samples in 2 different symbol sizes[6] and repeated the same operation using another identical set of data[7]. Since some 2D-barcodes only encode URL, we used 2 different URLs as our test data: a short and a long URL. This is done to see how data density and symbol size affect the FRR of each barcode.

Each barcode sample was captured by 2 different camera phones: Nokia 6600 with a VGA camera and Nokia 6630 with 1.3 megapixels camera. This allows us to observe the relationship between FRR and camera resolution.

Utilizing Cold Cathode Fluorescent lights, we captured each sample image from the most appropriate distance[8] (between 5 and 25 cm away from the target) under three different lighting conditions: lighting on full power, half power[9] and no additional lighting (see Fig 2). The room used for the experiments was lit by fluorescent ceiling lights. Hence, ambient light still existed when no additional lighting is used. The experiments assume a user performing the symbol capture process, hence, a certain amount of tilt/rotation of the captured images is expected.

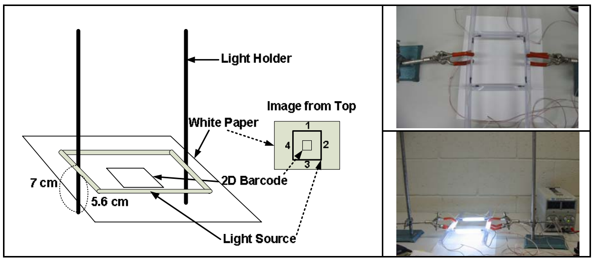

Figure 2. Light setting with top view (above) and front view (below).

[4] Error Checking and Correcting (ECC) 000 (0%), ECC 50 (2.8%), ECC 080(5.5%), ECC 100 (12.6%), ECC 140 (25%).

[5] A small amount of data such as phone numbers can be encoded and decoded from the code itself with no network connectivity.

[6] 2.5 cm^2 and 5.0 cm^2

[7] The percentage of error correction for all the QR Code and VSCode samples is set at 15%.

[8] Appropriate distance is subject to the reading software used, symbol size, data density, etc.

[9] It was administered in two ways: first, using the lights horizontally placed against the sample (light source 1 and 3 in Fig. 2) and then followed by those vertically placed (light source 2 and 4).

We calculated the FRRs by the numbers of successful first reads out of the number of attempts (i.e. 50 times).

More than one readers were available for some 2D-barcodes. We conducted the FRR experiments with all the available readers.[10] It should be noted that we used a personal computer (PC) based reader to decode VSCode symbols that are captured by the Nokia phones due to the unavailability of VSCode readers for camera phones.

IV. EXPERIMENT RESULTS AND OBSERVATIONS

A. Comparison between Index and database 2D-barcode

As far as FRRs are concerned, overall, index 2D-barcodes achieved better results (99.8%) than that of database 2D-barcodes (91.5%). With the exception of ColorCode FRRs (99.0%), the FRRs of the index 2D-barcodes were 100% regardless of lighting condition, symbol size, camera resolution, or data capacity (see Table I). However, the difference in the FRRs between the index and the database 2D-barcodes was insignificant. The FRRs of newer database 2D-barcode readers (i.e. Quick Mark, UpCode , mCode, and the Kaywa reader used for decoding Data Matrix) were also 100%. Furthermore, UpCode reader, which was able to read Data Matrix as well as index UpCode, achieved 100% FRRs for both 2D-barcodes.

B. Effect of data density and symbol size on FRRs

Unlike the index 2D-barcodes, factors such as symbol size and data density had a great impact on the FRRs of database 2D-barcodes. Under the same conditions, the FRRs of larger symbols were always higher than that of smaller ones. Likewise, generally, the FRR of denser symbols was lower than that of sparser symbols. This is because data density, together with symbol size, determines the cell size of each symbol. Bigger symbol size with less data in a given print area means an increase in the cell size of a 2D-barcode symbol.

C. Relation between camera resolution and FRRs

Higher camera resolution was not very important to read both black/white and color barcodes. This clearly rebuked the myth that we need expensive, high resolution camera to read color symbols. In fact, VGA camera often performed better than the 1.3 megapixels camera in terms of FRR.

According to Kozaki and Nishii [12], higher resolution of cameras does not always mean that they can produce better quality in images. This is especially true when the charge-coupled device (CCD) image sensor is implemented using the interline CCD (IT-CCD) architecture presented in Fig. 3, which is a standard for the current digital mobile phone cameras. We shall not get into the details of IT-CCD architectures, but a brief explanation should be appropriate.

TABLE I

MEAN FRR (IN %) OF EACH SAMPLE 2D-BARCODE IN DIFFERENT SYMBOL SIZES, DATA DENSITY AND CAMERA RESOLUTIONS

Data Density	Symbol Size ($sq.cm^2$)	QR Code				VSCode®		Data Matrix						mCode	
		Kaywa reader		Quick Mark reader		PC reader		Kaywa reader		Semacode reader		UpCode reader		Built-in reader	
		VGA	1.3M	VGA	1.3M	VGA	1.3M	VGA	1.3M	VGA	1.3M	VGA	1.3M	VGA	1.3M
Low	2.5	100	100	100	100	88.7	77.3	100	100	97.3	97.3	100	100	100	100
	5.0	100	100	100	100	100	87.3	100	100	99.3	99.3	100	100	100	100
High	2.5	0	0	100	100	60	54.6	100	100	69.3	14	100	100	100	100
	5.0	100	100	100	100	95.3	100	100	100	93.3	92	100	100	100	100

Data Density	Symbol Size ($sq.cm^2$)	UpCode		Visual Code		ShotCode		ColorCode	
		Built-in reader		Built-in reader		Built-in reader		Built-in reader	
		VGA	1.3M	VGA	1.3M	VGA	1.3M	VGA	1.3M
Low	2.5	100	100	100	100	100	100	98	100
	5.0	100	100	100	100	100	100	98	100
High	2.5	100	100	100	100	100	100	98	100
	5.0	100	100	100	100	100	100	98	100

ColorCode	
ColorCode reader for Nokia 6630	
Java v 2.0.3	SymbianOS v 1.1.0
98	100
96.7	100
98.6	100
97.3	100

A pixel is the basic unit of a digitized image. If the camera resolution is 1.3 megapixels, the surface of the CCD is divided into 1,300,000 pixels. The source of digital data is "light." To create a digital image data, firstly the CCD converts light captured via the photo-diode to electric charges, which are then gathered and sent to an amplifier via the Vertical CCD (VCCD) and horizontal CCD (HCCD). That is, each pixel is divided into two parts in the IT-CCD architecture.

When the camera resolution is increased, for example, from VGA (640 × 480) to 1.3 megapixels (1280 × 960) without changing the size of the CCD, both the size of the photo-diode and VCCD of each pixel would become one fourth of their original sizes. The problem is that we cannot reduce the size of the VCCD as small as one fourth of its original size and hence, the size of the photo-diode suffers. The photo-diode becomes even smaller than one fourth of its original size, which in turn reduces its surface to capture light, thus, degrading the quality of the captured image. Increasing the camera resolution may enable pinpoint accuracies. But, this has negative effect on the efficient capture of light in the IT-CCD architecture. This may explains our observations that the performance of VGA camera was better than that of the 1.3 megapixels.

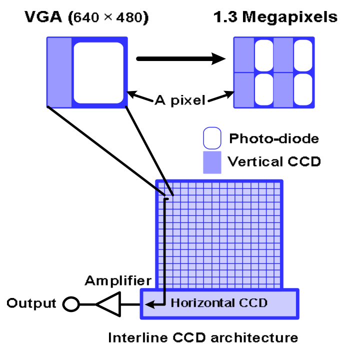

Figure 3. The effect on IT-CCD when resolution increases from VGA to 1.3 megapixels.

---

[10] VS Code reader was provided by Veritec Iconix Ventures Inc. (http://www.vi-vi.com/index.asp). Other publicly available readers were obtained from following sources: QuickMark < SimpleAct Inc.: http://www.quickmark.com.tw/English/download.html>, Kaywa <Kaywa™: http://reader.kaywa.com/>, Semacode <Semacode Corporation: http://semacode.org/about/hardware/>, mCode < Nextcode Corporation: http://www.connexto.com/Join.aspx>, Visual Code <ETH Zürich: http://www.inf.ethz.ch/personal/rohs/visualcodes/> , ShotCode <OP3: http://www.shotcode.com/download>, ColorCode < ColorZip SEA Pte Ltd. : http://www.colorcode.com.sg/Download.html >, UpCpde < UpCpde Ltd.: http://www.upc.fi/en/upcode/instructions/>

However, there are other factors we should consider to explain the difference in performance between Nokia 6600 and Nokia 6630 in addition to camera resolution: the version of reading software for each phone, differences in other camera features (e.g. auto-focus and sensitivity to lighting). Hence, while our results do not favor high resolution camera, neither do they imply that lower camera resolution is better.

Notably, Semacode reader for Nokia 6600 achieved five times better result than that for Nokia 6630 when reading a small dense Data Matrix symbol. In contrast, higher resolution camera always performed better in reading all the sampled ColorCodes. However, these results may be caused by the difference in programming platform rather than the difference in camera resolution. Available Semacode reader for Nokia 6630 and ColorCode reader for Nokia 6600 are programmed in Java, whereas the other ones are implemented in the native code (C++ in this case).[11] In our experiments, software written in C++ achieved considerably better results in terms of FRRs, legible distance, program execution speed, and their stability. In fact, the performance of the earlier ColorCode reader for Nokia 6630[12], which was implemented in Java, was not as high as ColorCode reader for Nokia 6600 in terms of both FRRs (see Table I) and legible distance. It took considerable time to get the knack and become able to read the target code with these readers. Moreover, the decoding time was around 30 seconds in average. This is rather long comparing with readers that are capable of immediate decoding. The current ColorCode reader for Nokia 6630 achieved doubled reading distance as well as 100% FRRs. Although Java's portability is appealing, this result indicates that careful consideration is required to choose a programming platform.

Similar to the FRRs, there were no significant differences in the maximum legible distances between the index and the database 2D-barcodes, except that data density has great effect on the latter. The general observation is simply, the bigger the symbol, the further a 2D-barcode can be successfully read.

Camera resolution had only a negligible effect on the legible distance of all the sampled 2D-barcodes except for the VSCode® symbols, which were decoded by the PC decoder.

D. Important findings

Three key factors to improve the robustness of 2D-barcode reading are the cell size of symbols, the decoding algorithm of the software and the reader hardware capability. For example, when using the Kaywa reader, the FRRs of the 2.5cm^2 QR Code with dense data were 0% regardless of camera resolution, whereas those of Data Matrix were 100%. The difference between them was the cells size. Once cells of a 2D-barcode become smaller than the recognizable size of a particular reader, the code cannot be successfully read.

The Quick Mark reader was superior to Kaywa reader in terms of FRR. However the reverse was true when it comes to code legible distance. Such differences in reading capability should result from variations in the reading software algorithm. The strong and continuous tracking ability of the Visual Code also shows that the reader software does make a difference.

The performance of decoding algorithms can be improved by better hardware capability. For example, with the zooming function of Nokia 6630 (i.e. 6x digital zoom), the Quick Mark reader can improve its reading distance up to 6 times, while up to 2 times when using the Nokia 6600 (i.e. 2x digital zoom).

V. Conclusions

Using Nokia 6600 (VGA camera) and Nokia 6630 (1.3 megapixels camera), we conducted first read rate examination of all the available and accessible 2D-barcodes used for camera phone applications. Since both 2D-barcode and camera phone technologies are rapidly improving, the current results could be replaced with new ones before long. Moreover, different implementation may results in different outcomes. Through our analysis, however, we identified three key factors to improve the reading robustness of 2D-barcodes, which are consistent and independent from the particular implementation. We also provided our observation on the programming platform. These informative findings can help researchers further improve the robustness of 2D-barcode reading, the range of the reading distance and the stability of the programs. It, in turn, enables to develop wider range of applications, hence, improve user experience. This could result in widespread use of 2D-barcode mobile technology as a ubiquitous computing tool.

References

[1] H. Kato and K. T. Tan, "2D Barcodes for Mobile Phones", IEE Mobility Conf. China, Page(s): P1A-4. November, 2005.

[2] H. Kato, "2D-barcode for Mobile Devices", Honours Thesis (1st Class), Dept. Computer and Information Science, Edith Cowan Univ., 2005.

[3] International Organization for Standardization: Information technology – Automatic identification and data capture techniques – Bar code symbology – QR Code. ISO/IEC 18004, 2000.

[4] VERITEC, Inc. (n.d.). "Benefits of Veritec's 2D Codes - VeriCode® and VSCode May 05, 2005; http://www.veritecinc.com/vericode.html.

[5] International Organization for Standardization: Information technology – International symbology specification – Data Matrix. ISO/IEC 16022, 2000.

[6] S.R.Sanford et al., "Efficient finder patterns and methods for application to 2D machine vision problems," US Patent 20060269136, US Patent and Trademark Office, November 30, 2006.

[7] M. Rohs, "Real-World Interaction with Camera-Phones," Proc.2nd Int'l Symp. Ubiquitous Computing Systems (UCS 04), LNCS 3598, Springer–Verlag, 2005, pp.74–89.

[8] A. Madhavapeddy et al., "Using Camera Phones to Enhance Human-Computer Interaction," Adjunct Proc.6th Int'l Conf. Ubiquitous Computing, 2004, http://ubicomp.org/ubicomp2004/adjunct/demos/madhavapeddy.pdf.

[9] T-D, Han, et. al., "Machine readable code image and method of encoding and decoding the same," US Patent 7,020,327, US Patent and Trademark Office, March 28, 2006.

[10] Semacode.org. "All the technical details you could possibly want and more," June 13, 2005; http://semacode.org/about/technical/.

[11] UPC Consulting Ltd./UpCode Ltd. "UpCode," March 25, 2007; http://www.upc.fi/en/upcode/.

[12] Y. Kozaki and Y. Nishii, "Study of digital camera mechanism," [体系的に学びなおす*デジタルカメラのしくみ*].Tokyo: NikkeiBP Soft Press, 2004.

---

[11] Semacode readers: Nokia 6600 - Semacode Reader Standalone for Series 60 Smartphones 1.5 (SymbianOS™, v 1.5.0), Nokia 6630 - Semacode Reader Standalone for Java Phones 1.6 (Java™, v1.6.0).
ColorCode readers: Nokia 6600 - ColorCode(Java™, v 0.0.0), Nokia 6630 - ColorCam (SymbianOS™, v 1.1.0).

[12] ColorCam (Java™, v 2.0.3)